

Measures of Nondeterminism for Pushdown Automata*

KAI SALOMAA

Department of Mathematics, University of Turku, SF-20500 Turku, Finland

AND

SHENG YU

*Department of Computer Science, University of Western Ontario,
London, Ontario, Canada N6A 5B7*

Received June 17, 1991; revised September 16, 1993

D. Vermeir and W. Savitch (*Fund. Inform.* 4 (1981), 401–418) introduced two measures of nondeterminism for pushdown automata and showed interestingly that the second measure, which we refer to as the depth measure, yields an infinite hierarchy of language families between the deterministic context-free and general context-free languages. However, the proof given in *op. cit.* for this hierarchy theorem was incorrect. In this paper, using a pumping result for deterministic context-free languages we give a new proof for the strictness of the depth hierarchy. We introduce the monadic depth measure which is also shown to give rise to an infinite hierarchy of language families. Furthermore, we show that the monadic hierarchy is shifted by at most one level from the unrestricted depth hierarchy. © 1994 Academic Press, Inc.

1. INTRODUCTION

Nondeterminism plays an important role in automata theory. For some machine models, such as pushdown automata, nondeterminism strictly enlarges the family of languages defined by the corresponding deterministic model. Nondeterministic finite automata recognize exactly the same family of languages as deterministic finite automata. However, the number of states needed to define a given language can be reduced exponentially by allowing the automaton to be nondeterministic. There are several important open problems concerning the equivalence of certain deterministic and nondeterministic resource bounded Turing machines, the most famous being the P and NP problem.

The study of Turing machines using a restricted amount of nondeterminism was initiated in [3]. Finite automata with a bounded amount of nondeterminism are considered in [1, 4]. Vermeir and Savitch introduced in [8] two measures of

* Research was supported by the Natural Sciences and Engineering Research Council of Canada grants OGP0041630 and by the Academy of Finland. A preliminary version of this paper appeared in the Proceedings of the 8th International Conference on Fundamentals of Computation Theory, Gosen, Germany, 1991.

nondeterminism for pushdown automata. The first measure counts the maximal number of nondeterministic steps in a computation on an input of a given length. This measure yields a hierarchy of only three levels. The second measure, where a nondeterministic pushdown automaton is viewed as a composition of deterministic components connected with a DAG (directed acyclic graph) structure, counts the depth of the DAG. The second measure will here be called the (nondeterministic) depth measure.

In this paper we show that the depth measure yields an infinite hierarchy of language families between the deterministic context-free languages and the context-free languages. This result was first stated in [8] but its proof was seriously flawed. In fact, all the languages used in [8] to separate the different families of the hierarchy belong to the first level above the deterministic languages. We will verify this in the paper. Our proof for the hierarchy result relies on a pumping lemma for deterministic context-free languages from [9]. Other pumping results for deterministic languages can be found in [2, 5]; see also [6].

Analogously to the depth measure one may define the size measure in terms of the number of components of a deterministic decomposition for a nondeterministic pushdown automaton. It turns out that languages of size k are exactly the languages of depth k , $k \geq 1$. Thus for a pushdown automaton with an arbitrary decomposition there always exists an equivalent automaton having the same depth where the corresponding DAG is linear. The result is in fact implicitly contained also in [8].

In a monadic decomposition the automaton is required always to have at most one nondeterministic move to another deterministic component. We show that also the monadic depth (and size) measure yields an infinite hierarchy. Furthermore, all languages of (unrestricted) depth k are shown to be of monadic depth $k + 1$. In other words, one can replace an arbitrary decomposition of depth k with a monadic decomposition of depth $k + 1$.

2. PRELIMINARIES

The reader is assumed to be familiar with the basics of formal language theory and with (deterministic) context-free languages in particular, cf., e.g., [2, 7]. Here we briefly introduce our notations and recall some of the definitions and results which are essential to this paper.

Let A and B be sets. We use the notation $A \subseteq B$ ($A \subset B$) to denote that A is a (proper) subset of B . The power set of A is denoted 2^A and if A is finite its cardinality is denoted $\#A$. Let Σ be a finite alphabet. The set of all (finite) words over Σ is denoted Σ^* , the length of a word $w \in \Sigma^*$ by $|w|$, and the empty word by λ . The reversal of w is denoted w^R . Let $A \subseteq \Sigma$. Then $\#_A(w)$, $w \in \Sigma^*$, denotes the number of occurrences of symbols of A in the word w . We define Er_A to be the homomorphism $\Sigma^* \rightarrow (\Sigma - A)^*$ erasing the symbols of A , i.e., Er_A is determined by $Er_A(a) = a$ if $a \in \Sigma - A$ and $Er_A(a) = \lambda$ for each $a \in A$.

For $w \in \Sigma^*$ we define $^{(k)}w$ to be the prefix of w of length k , $k \geq 1$,

$$^{(k)}w = \begin{cases} x, & \text{if } |w| > k, w = xy, |x| = k; \\ w, & \text{if } |w| \leq k. \end{cases}$$

Let $L_1, L_2 \subseteq \Sigma^*$. The catenation of the languages L_1 and L_2 is denoted $L_1 L_2$ and the right quotient of L_1 by L_2 is denoted L_1/L_2 and defined by

$$L_1/L_2 = \{u \in \Sigma^* \mid (\exists v \in L_2) uv \in L_1\}.$$

When there is no confusion, a singleton language $\{w\}$, $w \in \Sigma^*$, is denoted simply by w .

A pushdown automaton (pda) is a seven-tuple

$$A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

where Q is the finite set of states, Σ is the input alphabet, Γ is the pushdown alphabet,

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$$

is the transition relation defining the moves of the automaton where all values of δ are finite subsets of $Q \times \Gamma^*$, $q_0 \in Q$ is the initial state, $Z_0 \in \Gamma$ is the initial pushdown symbol, and $F \subseteq Q$ is the set of final states.

A pda A is said to be *deterministic* (dpda) if the following two conditions hold:

- (i) For all $(q, a, Z) \in Q \times (\Sigma \cup \{\lambda\}) \times \Gamma$, $\# \delta(q, a, Z) \leq 1$.
- (ii) Let $q \in Q$ and $Z \in \Gamma$. If $\delta(q, \lambda, Z) \neq \emptyset$, then $\delta(q, a, Z) = \emptyset$ for all $a \in \Sigma$.

The set of *instantaneous descriptions* of A is $ID(A) = Q \times \Sigma^* \times \Gamma^*$. The relation δ determines the relation $\vdash_A \subseteq ID(A) \times ID(A)$ representing the computations of A . The *language recognized by A* (by final state) is

$$L(A) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_A^* (q, \lambda, \alpha), q \in F, \alpha \in \Gamma^*\}.$$

A language L is a (deterministic) context-free language if there exists a (deterministic) pda A such that $L = L(A)$. The family of (deterministic) context-free languages over Σ is denoted (DCFL(Σ)) CFL(Σ). If Σ is arbitrary, we use the notation (DCFL) CFL. Finally, we recall the pumping lemma for deterministic languages from [9].

LEMMA 2.1 (Pumping Lemma). *Let L be a DCFL. Then there exists a constant C for L such that for any pair of words $w, w' \in L$ if*

- (i) $w = xy$ and $w' = xz$, $|x| > C$, and
- (ii) $^{(1)}y = ^{(1)}z$, then either (iii) or (iv) holds:

(iii) *there is a factorization $x = x_1 x_2 x_3 x_4 x_5$, $|x_2 x_4| \geq 1$ and $|x_2 x_3 x_4| \leq C$, such that, for all $i \geq 0$, $x_1 x_2^i x_3 x_4^i x_5 y$ and $x_1 x_2^i x_3 x_4^i x_5 z$ are in L ;*

(iv) *there exist factorizations $x = x_1 x_2 x_3$, $y = y_1 y_2 y_3$, and $z = z_1 z_2 z_3$, $|x_2| \geq 1$ and $|x_2 x_3| \leq C$, such that, for all $i \geq 0$, $x_1 x_2^i x_3 y_1 y_2^i y_3$ and $x_1 x_2^i x_3 z_1 z_2^i z_3$ are in L .*

3. MEASURES OF NONDETERMINISM

The nondeterminism degree (the depth measure) of a pushdown automaton was introduced in [8]. In the following, $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ denotes always a given pushdown automaton.

DEFINITION 3.1. Let $\emptyset \neq R \subseteq Q$ and define

$$\delta_R \subseteq (R \times (\Sigma \cup \{\lambda\}) \times \Gamma) \times 2^{R \times \Gamma^*}$$

as follows. For $a \in \Sigma \cup \{\lambda\}$ and $Z \in \Gamma$, $\delta_R(q, a, Z)$ is defined only if $q \in R$ and then

$$\delta_R(q, a, Z) = \{(r, \alpha) \in \delta(q, a, Z) \mid r \in R\}.$$

We say that R is a deterministic subset of Q (or of A) if (i) and (ii) below hold:

(i) For all $r \in R$ the pushdown automaton $(R, \Sigma, \Gamma, \delta_R, r, Z_0, F \cap R)$ is deterministic.

(ii) Let $r \in R$ and $Z \in \Gamma$. If $\delta_R(r, \lambda, Z) \neq \emptyset$ then $\#\delta(r, \lambda, Z) = 1$ and $\delta(r, a, Z) = \emptyset$ for all $a \in \Sigma$.

A deterministic subset R is said to be *monadic* if additionally the conditions (iii) and (iv) below hold. Let $q \in R$ and $Z \in \Gamma$.

(iii) For all $a \in \Sigma$, $\#[\delta(q, a, Z) - \delta_R(q, a, Z)] \leq 1$.

(iv) $\#\delta(q, \lambda, Z) \leq 1$ and if $\delta(q, \lambda, Z) \neq \emptyset$, then for all $a \in \Sigma$, $\delta(q, a, Z) - \delta_R(q, a, Z) = \emptyset$.

Intuitively, if R is a deterministic subset, then when in a state $r \in R$ (with a given input symbol and a given top-of-stack symbol), the automaton A has at most one transition leading to a state in R . Furthermore, if the automaton can make a λ -move to a state in R , then it has no other moves. In general, there can exist an arbitrary number of nondeterministic moves from $r \in R$ to a state not belonging to R but if R is a monadic deterministic subset, then the number of such moves is at most one.

Condition (ii) is not included in the corresponding definition of [8]. However, it is used there implicitly in the proof, showing that languages of depth k are exactly the k -repairable languages; see Theorem 3.3 below. (Without condition (ii) the pda M' constructed in the proof of Lemma 2 of [8] is not necessarily deterministic.)

DEFINITION 3.2. Let R_1 and R_2 be disjoint subsets of Q . We say that R_1 *directly precedes* R_2 , denoted $R_1 >_{dp} R_2$, if there exist $q_1 \in R_1, q_2 \in R_2, \alpha_1, \alpha_2 \in \Sigma^*$, and $\beta_1, \beta_2 \in \Gamma^*$, such that

$$(q_1, \alpha_1, \beta_1) \vdash_A^* (q_2, \alpha_2, \beta_2).$$

Let $\mathcal{R} = \{R_0, R_1, \dots, R_t\}$ be a partition of the state set Q . We say that the partition \mathcal{R} is *compatible* with the relation $>_{dp}$ if $>_{dp}$ defines a partial order on \mathcal{R} ; i.e., the transitive closure $>_{dp}^+$ of $>_{dp}$ is a partial order on \mathcal{R} .

Clearly, given $R_1, R_2 \subseteq Q$, it is decidable whether $R_1 >_{dp} R_2$. Also, one should note that the relation $>_{dp}$ is not, in general, transitive. Now we are ready to define the depth and size measures of a nondeterministic pda. The depth of a context-free language L is then naturally defined as the infimum of the depth measures of pushdown automata that recognize L .

DEFINITION 3.3. Let $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a pda. A *deterministic decomposition* of A is a partition $\mathcal{R} = \{R_0, R_1, \dots, R_t\}$ of Q , where each $R_i, i = 0, \dots, t$, is a deterministic subset and the partition \mathcal{R} is compatible with the relation $>_{dp}$. A deterministic decomposition $\mathcal{R} = \{R_0, R_1, \dots, R_t\}$ is said to be *monadic* if R_i is a deterministic monadic subset for all $i = 0, \dots, t$. If the longest chain in the partial order $>_{dp}^+$ has length $k + 1, k \geq 0$, then the decomposition \mathcal{R} is said to have *depth* k . The *size* of the above decomposition \mathcal{R} is defined to be t ; i.e., the size measure is the number of components minus one.

A pda A is said to be of nondeterministic depth (size) k if it has a deterministic decomposition of depth (size) k . A pda is of *finite nondeterministic depth (size)* if it is of depth (size) k for some $k \geq 0$. The family of languages recognized by nondeterministic pushdown automata having depth (size) k is denoted $\text{CFL}_{(k)}$ ($\text{CFL}^{(k)}$). Also, if $L \in \text{CFL}_{(k)}$ ($\text{CFL}^{(k)}$) we say that L is of depth (size) k . The family of languages of finite depth (size) is denoted by $\text{CFL}_{(\text{fin})}$ ($\text{CFL}^{(\text{fin})}$).

Analogously one defines the *monadic depth (size)* of a pda as the depth (size) of a monadic deterministic decomposition of the automaton. The family of languages recognized by nondeterministic pda's of monadic depth (size) k is denoted $\text{CFL}_{(k, 1)}$ ($\text{CFL}^{(k, 1)}$). The families of languages of finite monadic depth and size are denoted respectively $\text{CFL}_{(\text{fin}, 1)}$ and $\text{CFL}^{(\text{fin}, 1)}$.

Clearly $\text{CFL}_{(0)} = \text{CFL}_{(0, 1)} = \text{CFL}^{(0)} = \text{CFL}^{(0, 1)} = \text{DCFL}$. Let $L = \{ww^R \mid w \in \{0, 1\}^*\}$. It is easy to see that $L \in \text{CFL}^{(1, 1)}$. A pda for L can have a monadic deterministic decomposition with two components. The first component pushes input symbols into the stack and the second compares the stack with the remaining input.

The depth of a pda is called the nondeterminism degree in [8]. It turns out that languages of depth k are, in fact, exactly the languages of size k . Thus, for every pda with a decomposition of depth k there exists an equivalent pda having a decomposition of depth k such that $>_{dp}^+$ is a linear order. Intuitively, one can say that an arbitrary deterministic decomposition can always be linearized. The result below follows from the constructions given in [8].

THEOREM 3.1. $\text{CFL}_{(k)} = \text{CFL}^{(k)}$ for all $k \geq 0$.

Proof. Let $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a pda with a deterministic decomposition $\mathcal{R} = \{R_0, R_1, \dots, R_t\}$ of depth k . One can construct an equivalent pda A' of size k , where, intuitively, A' consists of $k+1$ copies of A . Each copy has only the transitions of A within each deterministic component of \mathcal{R} and the nondeterministic transitions of A between different components are simulated in A' by corresponding transitions from the i th copy to the $(i+1)$ th copy, $i = 0, \dots, k-1$. Below we give an outline of the construction of the pda A' since the same construction is used later also to establish that the monadic depth and size measures are equivalent.

The state set of the pda A' is $\bigcup_{i=0}^k Q_i$, where $Q_i = \{q_i \mid q \in Q\}$, $i = 0, \dots, k$, and the transition relation is defined so that $Q_i >_{dp}^+ Q_j$ iff $i < j$. Denote by $\mathcal{R}_i = \{R_{0,i}, R_{1,i}, \dots, R_{t,i}\}$ the partition of Q_i corresponding to \mathcal{R} . Within each Q_i the pda A' has only the transitions of A within each component $R_{j,i}$, $j = 0, \dots, t$. Let $(q, \alpha) \in \delta(p, a, Z)$ (respectively $(q, \alpha) \in \delta(p, \lambda, Z)$) be a nondeterministic transition of A between different components, i.e., $p \in R_r$, $q \in R_s$, $R_r >_{dp}^+ R_s$. The transition relation δ' of A' is defined so that, for every $i \in \{0, \dots, k-1\}$, $\delta'(p_i, a, Z)$ (respectively $\delta'(p_i, \lambda, Z)$) contains a pair (q_{i+1}, α) . Thus each Q_i is a deterministic subset of A' and hence A' is of size k . Since an arbitrary computation of A may involve at most k transitions between components of \mathcal{R} , it follows also that $L(A) = L(A')$. ■

Note that $>_{dp}^+$ is not necessarily a linear order on a monadic deterministic decomposition because distinct states of a given component may have nondeterministic transitions to different components. However, it is easy to check that if $\mathcal{R} = \{R_0, R_1, \dots, R_t\}$ is a monadic decomposition of the automaton A , then also the partition $\{Q_0, Q_1, \dots, Q_k\}$ of A' constructed in the proof of Theorem 3.1 is monadic. Thus we have

THEOREM 3.2. $\text{CFL}_{(k,1)} = \text{CFL}^{(k,1)}$ for all $k \geq 0$.

DEFINITION 3.4. Let $L \in \text{CFL}(\Sigma)$ and $k \geq 0$. The language L is said to be k -repairable if for some alphabet Δ , $\Delta \cap \Sigma = \emptyset$, there exists a language $L(k) \in \text{DCFL}(\Sigma \cup \Delta)$ such that $Er_\Delta(L(k)) = L$ and for all $w \in L(k)$, $\#_\Delta(w) \leq k$. The language L is said to be $(k, 1)$ -repairable if $L(k) \in \text{DCFL}(\Sigma \cup \Delta)$, where $\#_\Delta = 1$.

A language $L(k)$ as in the above definition is said to be a k -repairing language for L . The following characterization was given in [8]:

THEOREM 3.3. For every $k \geq 0$, $L \in \text{CFL}_{(k)}$ if and only if L is k -repairable.

The intuitive idea of the proof can be explained as follows. Assuming that L is accepted by a pda having a decomposition of depth k , one defines a k -repairing language $L(k)$ by inserting in the words of L markers that instruct the automaton to simulate a nondeterministic move between different components. If the current component is R_1 , the marker specifies the component R_2 to be entered, $R_1 >_{dp} R_2$, and whether this is done using a λ -move or when reading the next input symbol. By (i) and (ii) of Definition 3.1 it is easy to see that the repairing language can be

made deterministic. Conversely, a pda of depth k can simulate the computations of a dpda on a k -repairing language simply by guessing the positions where the markers are in the input. For the technical details see [8]. However, the proof given there seems to be unnecessarily complicated.

THEOREM 3.4. *For every $k \geq 0$, $L \in \text{CFL}_{(k,1)}$ if and only if L is $(k, 1)$ -repairable.*

Proof. The proof is essentially analogous to the proof of the previous theorem in [8], and we omit here the formal construction. In the case of a monadic decomposition, in any instantaneous description the pda has at most one nondeterministic move (a λ -move or a move reading the next input symbol) from the present deterministic component to some other component. Also, if such a move exists then there is no λ -move within the current component. The unique nondeterministic move can be simulated by a dpda if the input contains the unique marker of Δ before the next symbol of Σ . The simulation in the opposite direction is at least as straightforward. ■

In [8], it was stated that the language family $\text{CFL}_{(k)}$ is properly contained in the language family $\text{CFL}_{(k+1)}$ for all $k \geq 0$ and, thus, the families $\text{CFL}_{(k)}$ form an infinite hierarchy between the deterministic and the general context-free languages. However, the proof given in [8] is incorrect. In the following, we show that the languages used in [8] to separate the classes $\text{CFL}_{(k)}$ and $\text{CFL}_{(k+1)}$, $k \geq 0$, are all in $\text{CFL}_{(1)}$. Denote $L_1 = \{a^i b^j c^j \mid i, j \geq 1\}$ and $L_2 = \{a^i b^j c^j \mid i, j \geq 1\}$. Let

$$H_k = \$((L_1 \cup L_2) \$)^k, \quad k \geq 0.$$

It was claimed in [8] that $H_{k+1} \notin \text{CFL}_{(k)}$. The following disproves this claim.

CLAIM 3.1. *H_k is one-repairable for all $k \geq 1$.*

Proof. Let $\Sigma = \{a, b, c, \$ \}$. Choose a set Δ of cardinality 2^k and construct $H_k(1) \subseteq (\Sigma \cup \Delta)^*$ from H_k by adding a symbol of Δ at the beginning of each word indicating which of the following k components belong to L_1 and which to L_2 . Clearly $H_k(1)$ is deterministic and so H_k is one-repairable. Thus by Theorem 3.3, $H_k \in \text{CFL}_{(1)}$. ■

In fact, it is not difficult to see that $H_k \in \text{CFL}_{(1,1)}$ for all $k \geq 0$. One can construct a $(1, 1)$ -repairing language $H_k(1, 1) \in \text{DCFL}(\Sigma \cup \{d\})$ as follows. The position of d in each word of $H_k(1, 1)$ with length at least 2^k codes the information whether each of the k components of the word belongs to L_1 or L_2 . Words of length less than 2^k can be handled deterministically using only the finite-state memory.

4. THE DEPTH HIERARCHY

Here we give a proof for the hierarchy theorem of nondeterministic context-free languages. It relies on the characterization of depth k languages in terms of

k -repairable languages and on the pumping lemma for deterministic languages from [9].

THEOREM 4.1. $\text{DCFL} = \text{CFL}_{(0)} \subset \text{CFL}_{(1)} \subset \dots \subset \text{CFL}_{(k)} \subset \text{CFL}_{(k+1)} \subset \dots \subset \text{CFL}_{(\text{fin})} \subset \text{CFL}$. (Note that \subset denotes the proper-subset relation.)

Proof. Choose $\Sigma = \{0, 1\}$ and denote

$$L_0 = \{ww^R \mid w \in \Sigma^*\}.$$

For $k \geq 1$ define

$$M_k = (\$L_0)^k \$.$$

Clearly $M_{k+1} \in \text{CFL}_{(k+1)}$, $k \geq 0$. In order to show that $\text{CFL}_{(k)} \subset \text{CFL}_{(k+1)}$ it is sufficient to show that for all $k \geq 0$,

$$M_{k+1} \notin \text{CFL}_{(k)}.$$

Assume that $M_{k+1} \in \text{CFL}_{(k)}$. Let $U \in \text{DCFL}(\Omega)$ be a k -repairing language for M_{k+1} , where $\Omega = \Sigma \cup \{\$\} \cup \mathcal{A}$, $\mathcal{A} \cap (\Sigma \cup \{\$\}) = \emptyset$.

CLAIM 4.1. *There exist $r \in \{0, \dots, k\}$ and $w_1, \dots, w_r \in L_0$ such that the following condition holds:*

$$(\forall u \in L_0)(\exists v_i \in \text{Er}_{\mathcal{A}}^{-1}(w_i), i = 1, \dots, r) : \$v_1\$ \dots \$v_r\$u\$ \in U/\Omega^*. \quad (1)$$

Proof of the claim. Let $r \in \{0, \dots, k\}$. We say that $w_1, \dots, w_r \in L_0$ is an r -sequence for L_0 if for all $s \in \{1, \dots, r\}$ the following condition holds: For all $v_i \in \text{Er}_{\mathcal{A}}^{-1}(w_i)$, $i = 1, \dots, s-1$,

$$\$v_1\$ \dots \$v_{s-1}\$w_s\$ \notin U/\Omega^*.$$

Clearly there always exists a zero-sequence for L_0 . Assuming that we have found an r -sequence $w_1, \dots, w_r \in L_0$ for some r , $0 \leq r \leq k-1$, we claim that either the sequence w_1, \dots, w_r satisfies the condition (1) or there exists $w_{r+1} \in L_0$ such that w_1, \dots, w_r, w_{r+1} is an $(r+1)$ -sequence for L_0 . This follows immediately from the observation that the negation of condition (1) for w_1, \dots, w_r implies the existence of a word w_{r+1} such that w_1, \dots, w_r, w_{r+1} is an $(r+1)$ -sequence.

Finally, if w_1, \dots, w_k is a k -sequence for L_0 then (1) holds for w_1, \dots, w_k . This follows from the fact that each word of U contains at most k symbols from \mathcal{A} and every prefix of a word of U belonging to $\text{Er}_{\mathcal{A}}^{-1}(\$w_1\$ \dots \$w_k\$)$ contains at least k symbols from \mathcal{A} . This concludes the proof of the claim. ■

Now we proceed to derive a contradiction from the assumption that $U \in \text{DCFL}$. Let $w_1, \dots, w_r \in L_0$, $0 \leq r \leq k$, be words such that (1) holds. Deterministic languages are closed under intersection with regular sets and right-quotient with regular sets; cf. [2]. Hence for all words $v_i \in \text{Er}_{\mathcal{A}}^{-1}(w_i)$, $i = 1, \dots, r$, the language

$$L(v_1, \dots, v_r) = (U/\Omega^*) \cap \$v_1\$ \dots \$v_r\$\Sigma^*\$$$

is deterministic. Assume that

$$v_i \in Er_A^{-1}(w_i), i = 1, \dots, r \text{ and } L(v_1, \dots, v_r) \neq \emptyset. \quad (2)$$

Then clearly $\#_A(v_i) \leq k, i = 1, \dots, r$, and thus there exist only finitely many r -tuples v_1, \dots, v_r such that (2) holds.

For each nonempty language $L(v_1, \dots, v_r)$, $C(v_1, \dots, v_r)$ denotes the constant from Lemma 2.1 corresponding to $L(v_1, \dots, v_r)$. If $L(v_1, \dots, v_r) = \emptyset$ define $C(v_1, \dots, v_r) = 0$. Let

$$p = \max \{ C(v_1, \dots, v_r) \mid v_i \in Er_A^{-1}(w_i), i = 1, \dots, r \}.$$

For $n \geq 1$, denote

$$w(n) = (10^p 1)^n.$$

Note that $w(2n) \in L_0$ for all $n \geq 0$. Since there are only finitely many nonempty languages $L(v_1, \dots, v_r), v_i \in Er_A^{-1}(w_i), i = 1, \dots, r$, there exist words $u_i \in Er_A^{-1}(w_i), i = 1, \dots, r$, and $n_1, n_2 \geq 1, n_2 \geq 2n_1$ such that

$$\$u_1 \$ \cdots \$u_r \$w(2n_1) \$ \in L(u_1, \dots, u_r)$$

and

$$\$u_1 \$ \cdots \$u_r \$w(2n_2) \$ \in L(u_1, \dots, u_r).$$

Denote $u = \$u_1 \$ \cdots \u_r . In the notations of Lemma 2.1 we choose

$$x = u\$ (10^p 1)^{2n_1 - 1} 10^p,$$

$$y = 1\$,$$

$$z = 1(10^p 1)^{2(n_2 - n_1)} \$.$$

Thus xy and xz belong to the deterministic language $L(u_1, \dots, u_r), {}^{(1)}y = {}^{(1)}z$, and $|x| > C(u_1, \dots, u_r)$.

First assume that condition (iii) of Lemma 2.1 holds and let $x = x_1 x_2 x_3 x_4 x_5$ be an arbitrary decomposition such that $|x_2 x_4| \geq 1$ and $|x_2 x_3 x_4| \leq p$. Since $2n_1 \leq 2(n_2 - n_1)$, it is easy to verify that

$$x_1 x_3 x_5 z \notin L(u_1, \dots, u_r).$$

Second, assume that condition (iv) of Lemma 2.1 holds. Let $x = x_1 x_2 x_3$ be an arbitrary decomposition of x such that $x_2 \neq \lambda$ and $|x_2 x_3| \leq p$. Then $x_2 x_3 = 0^i$ for some $i, 1 \leq i \leq p$. Clearly for all decompositions $y_1 y_2 y_3$ of $y = 1\$$ the word $x_1 x_3 y_1 y_3$ is not in $L(u_1, \dots, u_r)$. By the pumping lemma, $L(u_1, \dots, u_r) \notin \text{DCFL}$.

Thus we have shown that $\text{CFL}_{(k)} \subset \text{CFL}_{(k+1)}$ for all $k \geq 0$. This implies also that $\text{CFL}_{(k)} \subset \text{CFL}_{(\text{fin})}$ for all $k \geq 0$. It remains to be proved that $\text{CFL}_{(\text{fin})} \subset \text{CFL}$.

Let L_0 be defined as above and denote

$$M = (\$L_0)^* \$ \in \text{CFL}.$$

Assume that $M \in \text{CFL}_{(\text{fin})}$, i.e., there exists an integer $k \geq 0$ such that $M \in \text{CFL}_{(k)}$. Let $M(k)$ be a k -repairing language for M over an alphabet $\Sigma \cup \{\$\} \cup \mathcal{A}$ such that $\mathcal{A} \cap (\Sigma \cup \{\$\}) = \emptyset$. Then

$$M'(k) = M(k) \cap (\$(\Sigma \cup \mathcal{A})^*)^{k+1} \$ \in \text{DCFL}$$

and

$$\text{Er}_{\mathcal{A}}(M'(k)) = M_{k+1}.$$

Hence, $M'(k)$ is a k -repairing language for M_{k+1} . This is a contradiction since it was shown above that $M_{k+1} \notin \text{CFL}_{(k)}$. This also completes the proof of Theorem 4.1. ■

Note that the language M in the preceding proof is unambiguous and the languages H_k in Claim 3.1 are inherently ambiguous. Thus each of the families $\text{CFL}_{(k)}$, $k \geq 1$, is incomparable with the family of unambiguous languages.

5. THE MONADIC HIERARCHY

Above it was observed that $\text{CFL}_{(k, 1)}$ consists of exactly all k -repairable languages where the "repairing alphabet" \mathcal{A} is restricted to have cardinality one. The languages M_{k+1} used to separate the families $\text{CFL}_{(k)}$ and $\text{CFL}_{(k+1)}$ in the proof of Theorem 4.1 clearly are $(k+1)$ -repairable using an alphabet of cardinality one. Thus we have immediately the following result stating that the monadic depth hierarchy is strict.

THEOREM 5.1. $\text{CFL}_{(k, 1)} \subset \text{CFL}_{(k+1, 1)}$ for all $k \geq 0$.

Next we compare the monadic hierarchy with the unrestricted depth hierarchy. As observed above, from the proof of Theorem 4.1 it follows that $\text{CFL}_{(k+1, 1)} - \text{CFL}_{(k)} \neq \emptyset$ for all $k \geq 0$. We will show that one can always replace k markers from an arbitrary alphabet with $k+1$ markers belonging to \mathcal{A} of cardinality one.

LEMMA 5.1. Let $L \in \text{CFL}(\Sigma)$ and $L(k) \in \text{DCFL}(\Sigma \cup \mathcal{A})$ be a k -repairing language for L , $k \geq 1$. Let Ω denote $\mathcal{A} \cup \mathcal{A}'$, where $\mathcal{A}' = \{\gamma' \mid \gamma \in \mathcal{A}\}$. Then for every $m \geq 1$ there exists a k -repairing language $L^{(m)}(k) \in \text{DCFL}(\Sigma \cup \Omega)$ for L such that

$${}^{(m)}w \in \Sigma^* \quad \text{for all } w \in L^{(m)}(k). \quad (3)$$

Proof. Assume that $m \geq 1$ is given. Let $w \in (\Sigma \cup \Omega)^*$. We say that a symbol $\omega \in \Omega$ occurs at the position i in w , $1 \leq i \leq |\text{Er}_{\Omega}(w)| + 1$, if ω appears between the

$(i-1)$ th and i th symbol of Σ in w . (Note that a word belonging to a k -repairing language may have up to k symbols of Ω at a given position.)

For each word $w \in L(k)$, we define $w(m) \in (\Sigma \cup \Omega)^*$ as follows:

(i) If $|Er_A(w)| < 2m$ then $w(m) = Er_A(w)$.

(ii) Suppose that $|Er_A(w)| \geq 2m$. Then $w(m)$ is obtained from w by removing each string of markers $\gamma_1 \cdots \gamma_r$ ($\gamma_j \in A$) appearing in a position i , $1 \leq i \leq m$, and adding a string $\gamma'_1 \cdots \gamma'_r$ at position $i+m$ after all symbols of A possibly appearing in the position $i+m$.

Then we define

$$L^{(m)}(k) = \{w(m) \mid w \in L(k)\}.$$

Clearly $Er_\Omega(L^{(m)}(k)) = Er_A(L(k)) = L$, $\#_\Omega(w) \leq k$ for all $w \in L^{(m)}(k)$, and no prefix of length m of any word in $L^{(m)}(k)$ contains a symbol of Ω .

Let A be a dpda such that $L(A) = L(k)$. As a modification of A we define a dpda A' recognizing the language $L^{(m)}(k)$. Given an input word $w \in (\Sigma \cup \Omega)^*$, where $|Er_\Omega(w)| < 2m$, using only the finite-state memory the automaton A' can check whether $w \in L^{(m)}(k)$. (Note that in this case $w \in L^{(m)}(k)$ iff $w \in L$.) Thus in the following we can consider only inputs w , where $|Er_\Omega(w)| \geq 2m$.

The pda A' always first reads from an input word the first $2m$ symbols of Σ and stores these and the possible symbols of Ω appearing in positions $i \leq 2m$ in its finite-state memory. (If it finds more than k symbols of Ω , A' rejects automatically.) After this, A' directly simulates the computation of A by considering a symbol $\gamma' \in A'$ in a position $i \in \{m+1, \dots, 2m\}$ to be the corresponding symbol $\gamma \in A$ in the position $i-m$. Clearly A' is deterministic. Thus $L^{(m)}(k)$ is a k -repairing language for L satisfying (3). ■

In the above proof the alphabet of markers Ω corresponding to $L^{(m)}(k)$ is larger than the original alphabet of markers A . However, Ω is independent of m . This fact will be needed in the following.

THEOREM 5.2. $\text{CFL}_{(k)} \subset \text{CFL}_{(k+1, 1)}$ for all $k \geq 0$.

Proof. By the proof of Theorem 4.1 it is sufficient to show that $\text{CFL}_{(k)} \subseteq \text{CFL}_{(k+1, 1)}$, $k \geq 0$. Let $L \in \text{CFL}(\Sigma)$ and $L(k) \in \text{DCFL}(\Sigma \cup A)$ be a k -repairing language for L . Let $d = \#A$ and choose

$$m = (2d)^{k+1}.$$

By Lemma 5.1, there exists for L a k -repairing language $L^{(m)}(k) \in \text{DCFL}(\Sigma \cup \Omega)$, where $\# \Omega = 2(\#A)$ such that ${}^{(m)}w \in \Sigma^*$ for all $w \in L^{(m)}(k)$.

Let $\Gamma = \{\gamma\}$ and define the homomorphism $h : (\Sigma \cup \Omega)^* \rightarrow (\Sigma \cup \Gamma)^*$ by

$$h(a) = \begin{cases} a, & \text{if } a \in \Sigma; \\ \gamma, & \text{if } a \in \Omega. \end{cases}$$

We denote

$$S = \{a_1 \cdots a_i \mid 0 \leq i \leq k, a_j \in \Omega, j = 1, \dots, i\}$$

and let $g : S \rightarrow \{1, \dots, m\}$ be an injective function. (Note that $\#S < m$.) Now corresponding to a word $w \in L^{(m)}(k)$ we define $w' \in (\Sigma \cup \Gamma)^*$ as follows.

If $\#_\Sigma(w) < m$ then choose $w' = h(w)$. Assume that $\#_\Sigma(w) \geq m$ and $g(Er_\Sigma(w)) = i$ ($i \in \{1, \dots, m\}$). Then w' is obtained from $h(w)$ by adding one symbol γ before the i th symbol of $h(w)$. Intuitively, the position of the additional symbol γ codes the labels of the at most k symbols of Ω that have all been replaced by the symbol γ .

We define $M \subseteq (\Sigma \cup \Gamma)^*$ by

$$M = \{w' \mid w \in L^{(m)}(k)\}.$$

Let A be a dpda such that $L(A) = L^{(m)}(k)$. Then the language M can be recognized by a dpda A' that counts the position i ($1 \leq i \leq m$) of the first symbol γ in the input, and otherwise A' simulates the behaviour of A assuming the remaining symbols γ to be replaced by the sequence $g^{-1}(i)$. (Again A' can handle words w with $\#_\Sigma(w) < m$ using only the finite-state memory.) Also, it is clear that for all $u \in M$, $\#_T(u) \leq k + 1$ and $Er_T(M) = L$. Thus $L \in \text{CFL}_{(k+1, 1)}$. ■

The results of Theorems 4.1, 5.1, and 5.2 are summarized in Fig. 1. In the figure, the arrows indicate strict inclusions and dashed arrows indicate inclusions that are not known to be strict.

The main open question concerning the interrelationships of the monadic and unrestricted depth hierarchies is whether $\text{CFL}_{(k, 1)} = \text{CFL}_{(k)}$, $k \geq 1$. Let M_1 be as in the proof of Theorem 4.1 and H_k as in Claim 3.1. At first one might suppose that languages like $M_1 H_k$ could belong to $\text{CFL}_{(1)} - \text{CFL}_{(1, 1)}$. However, a dpda can recognize the language M_1 if all words contain a single marker d at any distance of at most a fixed constant from the "middle" of the word. Thus the position of d modulo 2^k can code the necessary information about the suffix of the input belonging to H_k and, in fact, $M_1 H_k \in \text{CFL}_{(1, 1)}$. Using similar constructions, one can transform a k -repairing language to a $(k, 1)$ -repairing language in all of the examples that we have considered.

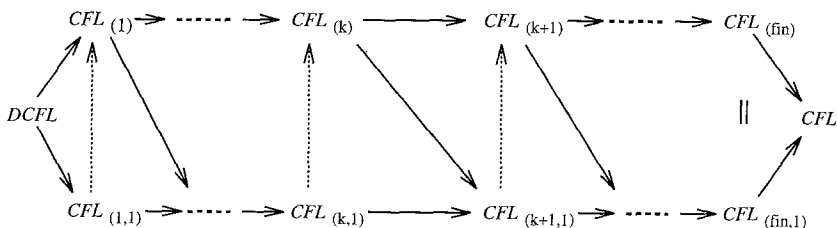


FIGURE 1

REFERENCES

1. J. GOLDSTINE, C. M. R. KINTALA, AND D. WOTSCHKE, On measuring nondeterminism in regular languages, *Inform. and Comput.* **86** (1990), 179–194.
2. M. A. HARRISON, “Introduction to Formal Language Theory,” Addison–Wesley, Reading, MA, 1978.
3. C. M. R. KINTALA AND P. C. FISCHER, Computations with a restricted number of nondeterministic steps, in “Proceedings, 9th ACM Symposium on the Theory of Computing, 1977,” pp. 178–185.
4. C. M. R. KINTALA AND D. WOTSCHKE, Amounts of nondeterminism in finite automata, *Acta Inform.* **13** (1980), 199–204.
5. M. LI AND P. VITANYI, A new approach to formal language theory by Kolmogorov complexity, in “Proceedings 16th ICALP,” Lect. Notes in Comput. Sci., Vol. 372, pp. 506–520, Springer-Verlag, New York/Berlin, 1989.
6. W. F. OGDEN, “Intercalation theorems for pushdown store and stack languages,” Ph.D. thesis, Stanford University, 1968.
7. A. SALOMAA, “Formal Languages,” Academic Press, New York, 1973.
8. D. VERMEIR AND W. J. SAVITCH, On the amount of nondeterminism in pushdown automata, *Fund. Inform.* **4** (1981), 401–418.
9. S. YU, A pumping lemma for deterministic context-free languages, *Inform. Process. Lett.* **31** (1989), 47–51.